**Algorithm**

1. Start the program execution in the main method of the Main class.
2. Create objects for the CameraOperations, UserInfo, and Scanner classes.
3. Initialize the user's username, password, and wallet amount.
4. Prompt the user to enter their username and password to log in.
5. Validate the user's credentials by calling the validateUser method.
6. If the login credentials are valid, display a success message and proceed to the process method.
7. In the process method, provide a menu of options to the user.
8. Depending on the user's choice, perform the corresponding actions:

* Option 1: My Camera
* Submenu: Add, Remove, View my Cameras, or Go to the previous menu.
* Option 2: Rent a Camera
* Show all available cameras.
* Prompt the user to enter the camera ID they want to rent.
* If the camera is available and the user has sufficient funds in their wallet, rent the camera.
* Deduct the rent amount from the user's wallet and display the rented camera details.
* Option 3: View All Cameras
* Display all the cameras along with their details.
* Option 4: My Wallet
* Ask the user if they want to add money to their wallet or not.
* Depending on the choice, either add money to the wallet or display the current wallet amount.
* Option 5: Exit
* Display a goodbye message and exit the program.

1. Handle invalid input by catching exceptions and providing appropriate error messages.
2. Repeat the menu prompt until the user chooses to exit the program.
3. The validateUser method checks if the entered username and password match the predefined values.
4. If the credentials are valid, call the process method.
5. If the credentials are invalid, display an error message and prompt the user to login again.
6. The login method prompts the user to enter their username and password.
7. Validate the input and call the validateUser method to check the credentials.
8. The CameraOperations class provides methods to add a camera, remove a camera, display all cameras, and rent a camera.
9. The UserInfo class represents user information, including username, password, and wallet amount.
10. The Camera class represents camera details, such as camera ID, brand, model, rent per day, and availability.
11. Utilize appropriate data structures like lists and variables to store and manipulate camera and user information.
12. Implement additional methods as required for specific functionalities.
13. Handle exceptions and errors gracefully by displaying error messages and providing options to retry or exit the program.
14. Ensure the program follows proper coding conventions and best practices.

**Github Link :** <https://github.com/PVRevanth2k/AssessmentProject.git>